Lisbon School
of Economics
& Management
Universidade de Lisboa

LISBOA
UNIVERSIDADE
DE LISBOA

Visualization

Prof. Carlos J. Costa, PhD

Learning Goals

- Know the main Python libraries used in data visualization
- Create line charts
- Create bar charts
- Create scatter plots
- Compare the data visualization libraries

matplotlib

matplotlib

- powerful and versatile Python library for creating static
- supports chart types, including scatter plots, line charts, and bar charts.
- While highly flexible, it can sometimes feel verbose for simple visualizations.
- it serves as the backbone for many other libraries, such as Seaborn and Plotly
- • **Strengths:**
  - Cntrol over plot elements
  - Wide range of plot types
  - Suitable for publication-quality figures
- • **Weaknesses:**
  - Learning curve for beginners
  - More verbose compared to higher-level libraries

- The building blocks of Matplotlib plots include:
  - Figure (fig)
  - Axes (ax)
  - Subplots
  - Axis

Figure (fig)

- is the top-level container that holds all plot elements
- Includes one or more subplots.
- Like a blank canvas where multiple charts can be arranged.

```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(8, 6))



fig = plt.figure(figsize=(5, 8))

ax = fig.add_subplot()

plt.show()
```

Axes (ax)

- Axes (ax) is the area where data is plotted.
- A figure can contain multiple axes (subplots), each displaying a different visualization.
  - `fig, ax = plt.subplots()`
  - `ax.plot([1, 2, 3], [4, 5, 6])`
  - `plt.show()`

Subplots

- Are multiple independent plotting areas within the same figure
- useful for comparing different datasets side by side

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 4))

ax1.plot([1, 2, 3], [4, 5, 6], label="Dataset 1")

ax2.scatter([1, 2, 3], [6, 5, 4], color='r', label="Dataset 2")

plt.show()
```

Axis

- corresponds to x-axis and y-axis,
- define the coordinate space for data visualization.
- Each axis can be labeled, scaled, and customized
    - `fig, ax = plt.subplots()`
    - `ax.plot([10, 20, 30], [5, 15, 25])`
    - `ax.set_xlabel("X-Axis Label")`
    - `ax.set_ylabel("Y-Axis Label")`
    - `ax.set_title("Example Plot with Labeled Axes")`
    - `plt.show()`

Import data

```
import pandas as pd
```

```
df = pd.read_csv(‘electionsUSA.csv’)
```

File:

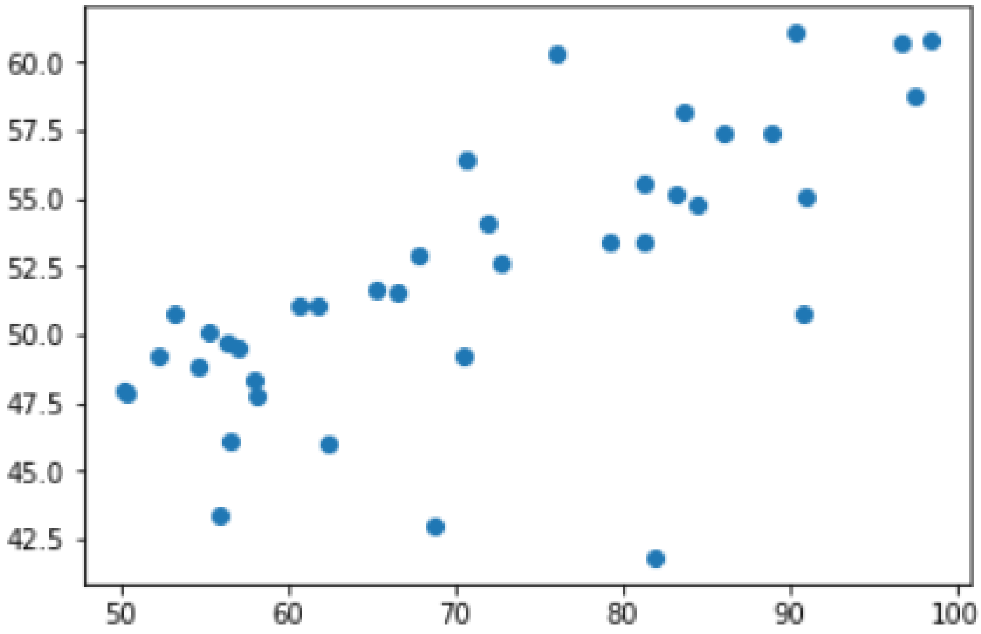
https://raw.githubusercontent.com/masterfloss/data/main/electionsUSA.csv

Import data

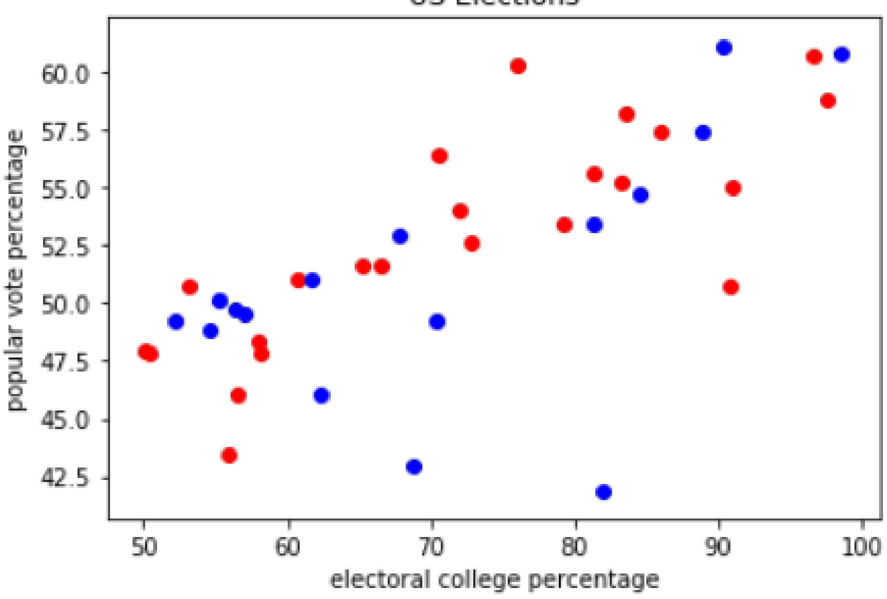
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('electionsUSA.csv')
```

Scatter Plot

```
# create a figure and axis
fig, ax = plt.subplots()
# scatter the 'electoral college
# percentage against the
# popular vote  percentage
ax.scatter(df['electoral college
percentage'], df['popular vote
percentage'])
```

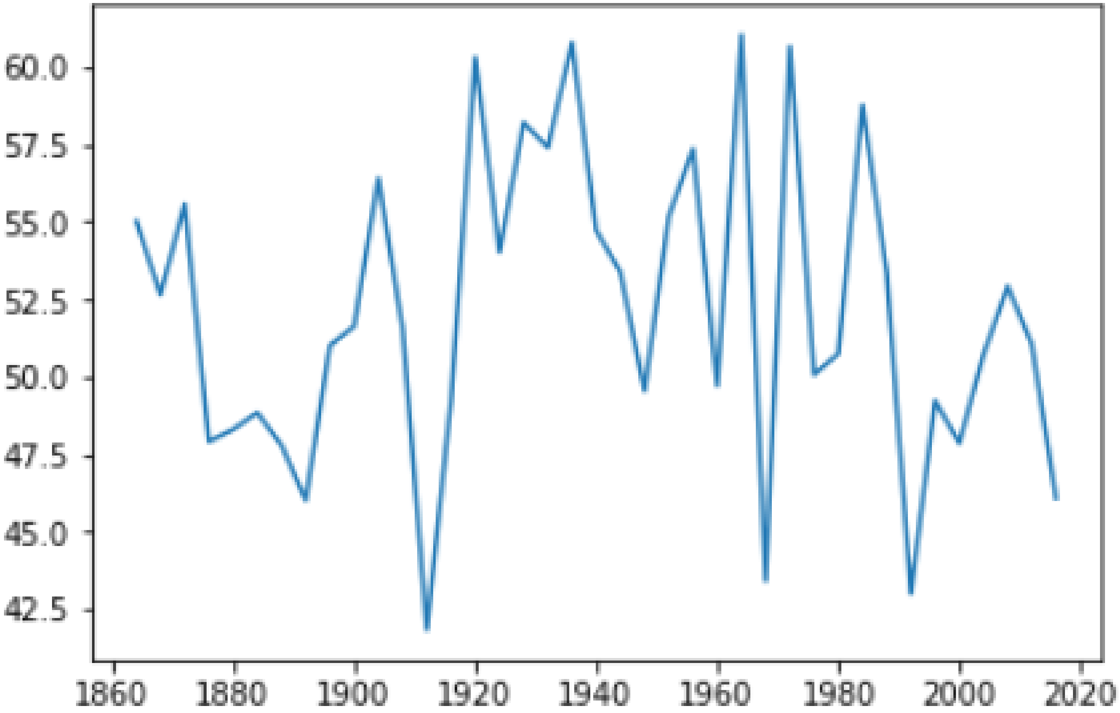


Scatter Plot



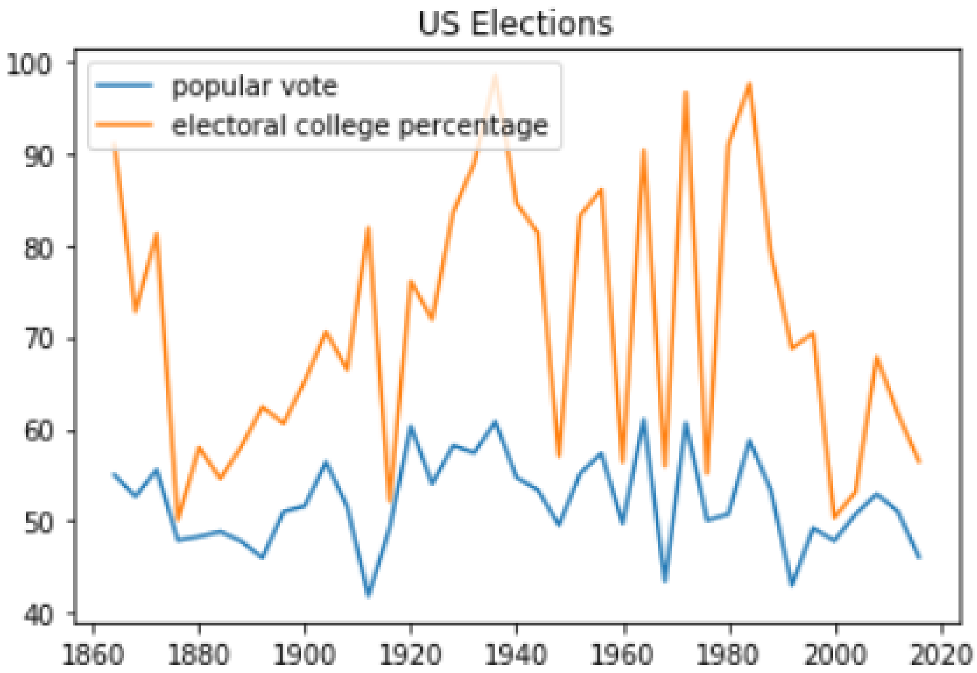
```
# create colour dictionary
colrs = {'Rep.':'r', 'Dem.':'b'}
# create a figure and axis
fig, ax = plt.subplots()
# plot each data-point
for i in range(len(df['electoral college percentage'])):
   ax.scatter(df['electoral college percentage'][i],df['popular vote percentage'][i],colr=colrs[df['party'][i]])
# set a title
ax.set_title('US Elections')
ax.set_xlabel('electoral college percentage')
ax.set_ylabel('popular vote percentage')
```

Line Chart



```
fig, ax = plt.subplots()
ax.plot(df['year'],df['popular vote percentage'])
```

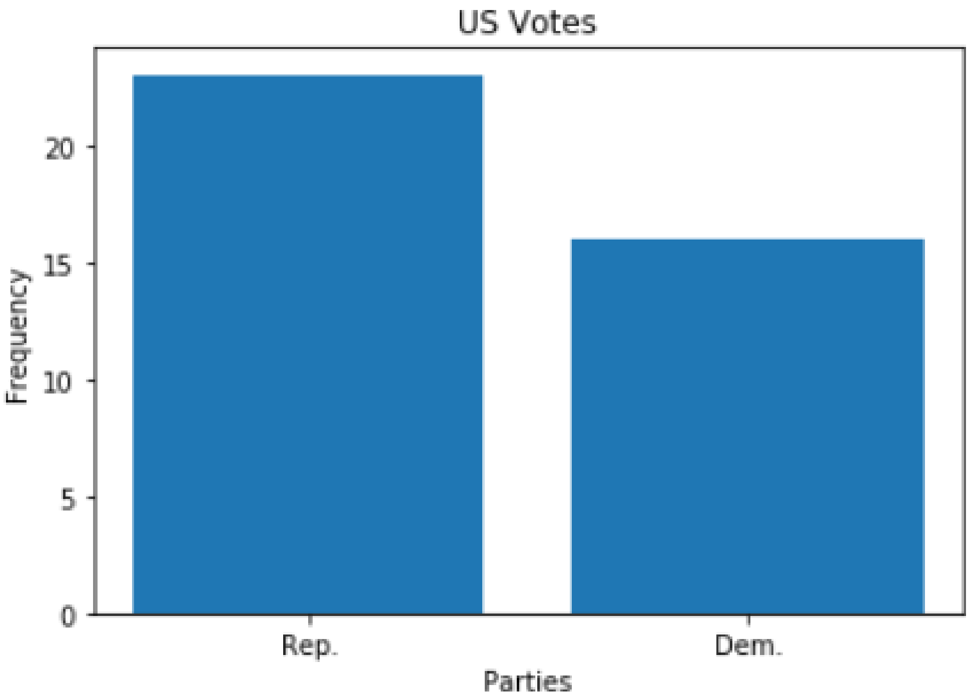
Line Chart



```
fig, ax = plt.subplots()

ax.plot(df['year'],df['popular vote percentage'],
label="popular vote")

ax.plot(df['year'],df['electoral college percentage'],
label="electoral college percentage")

ax.set_title('US Elections')

ax.legend()
```

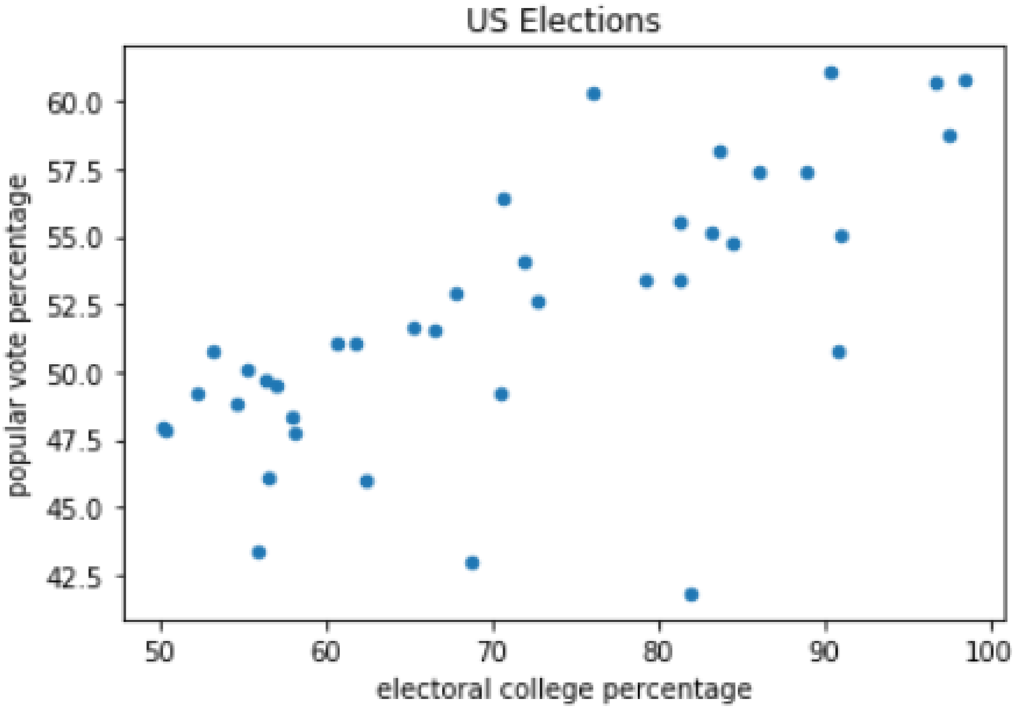
Bar Chart

```
# create a figure and axis
fig, ax = plt.subplots()
# count the occurrence of each class
data = df['party'].value_counts()
# get x and y data
points = data.index
frequency = data.values
# create bar chart
ax.bar(points, frequency)
# set title and labels
ax.set_title('US Votes')
ax.set_xlabel('Parties')
ax.set_ylabel('Frequency')
```



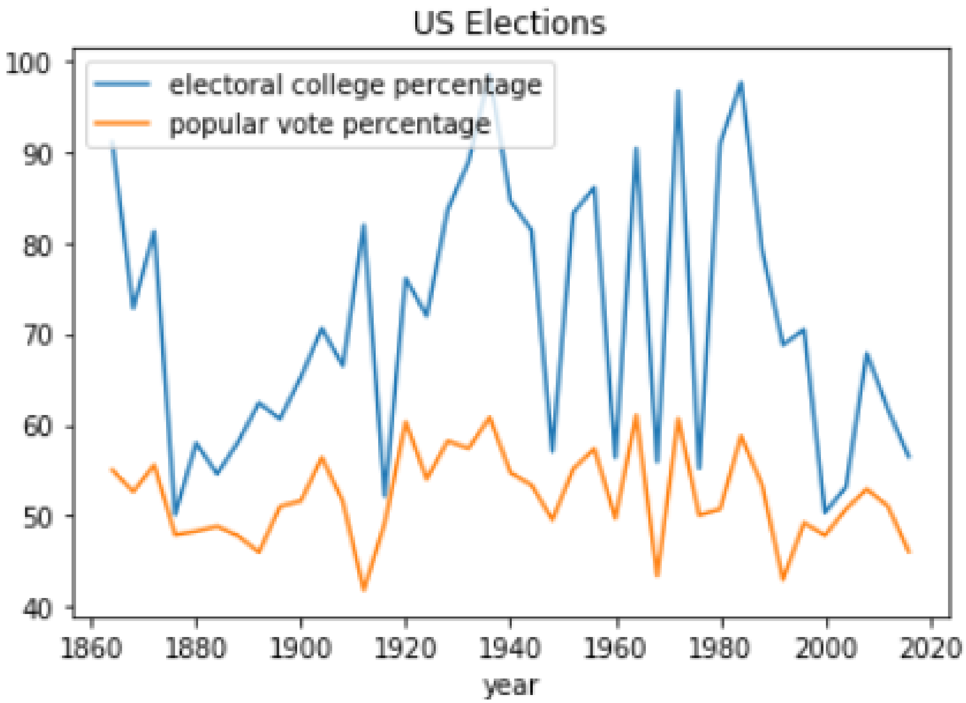
pandas

Bar Chart

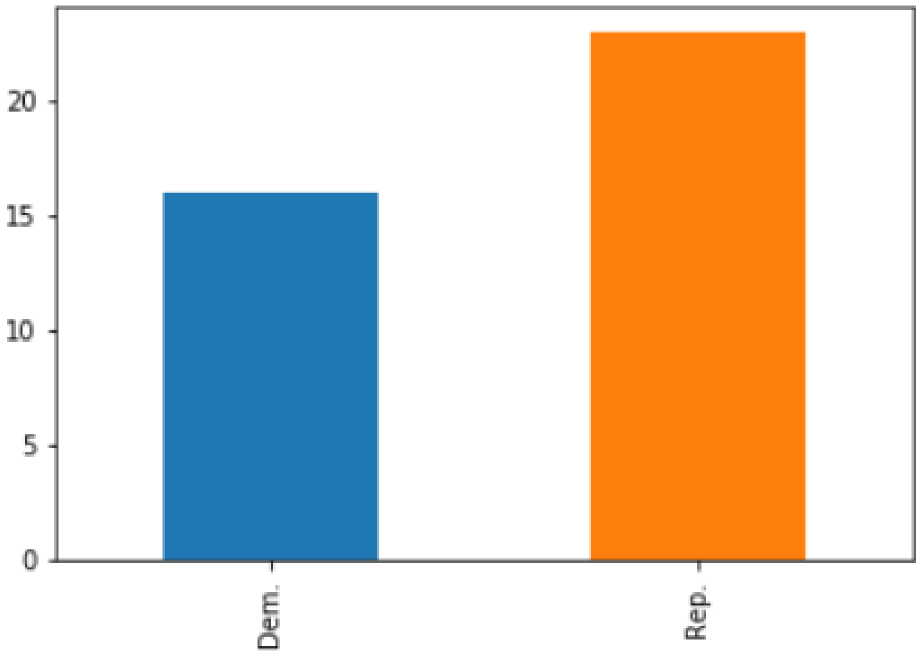


```
import pandas as pd
import matplotlib.pyplot as plt
url=df = pd.read_csv('electionsUSA.csv')
gr=df.plot.scatter(x='electoral college percentage', y='popular vote percentage', title='US Elections’)
gr.plot
```

Line Chart



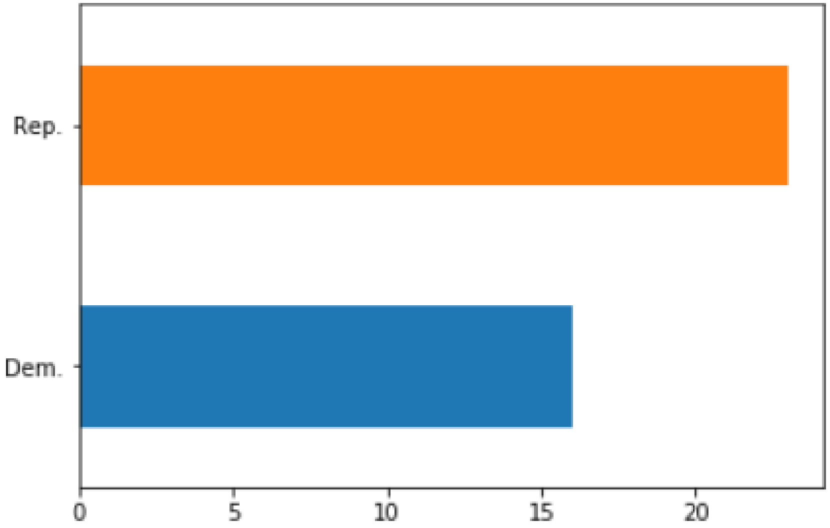
```
df=df.set_index('year')

gr=df.drop(['party'], axis=1).plot.line(title='US Elections')

gr.plot
```



Bar Chart

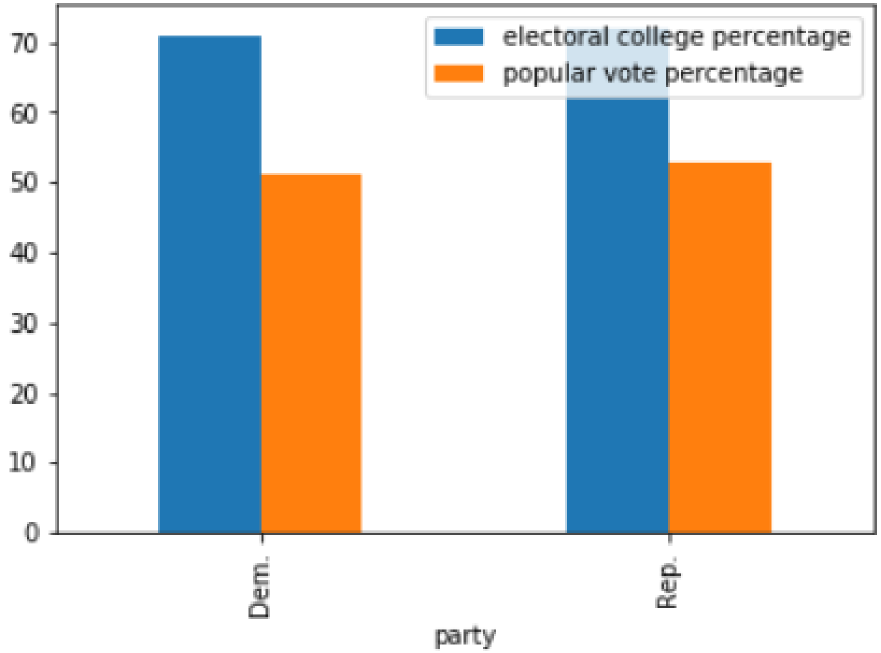
```
df['party'].value_counts().sort_index().plot.bar()
```

Bar Chart



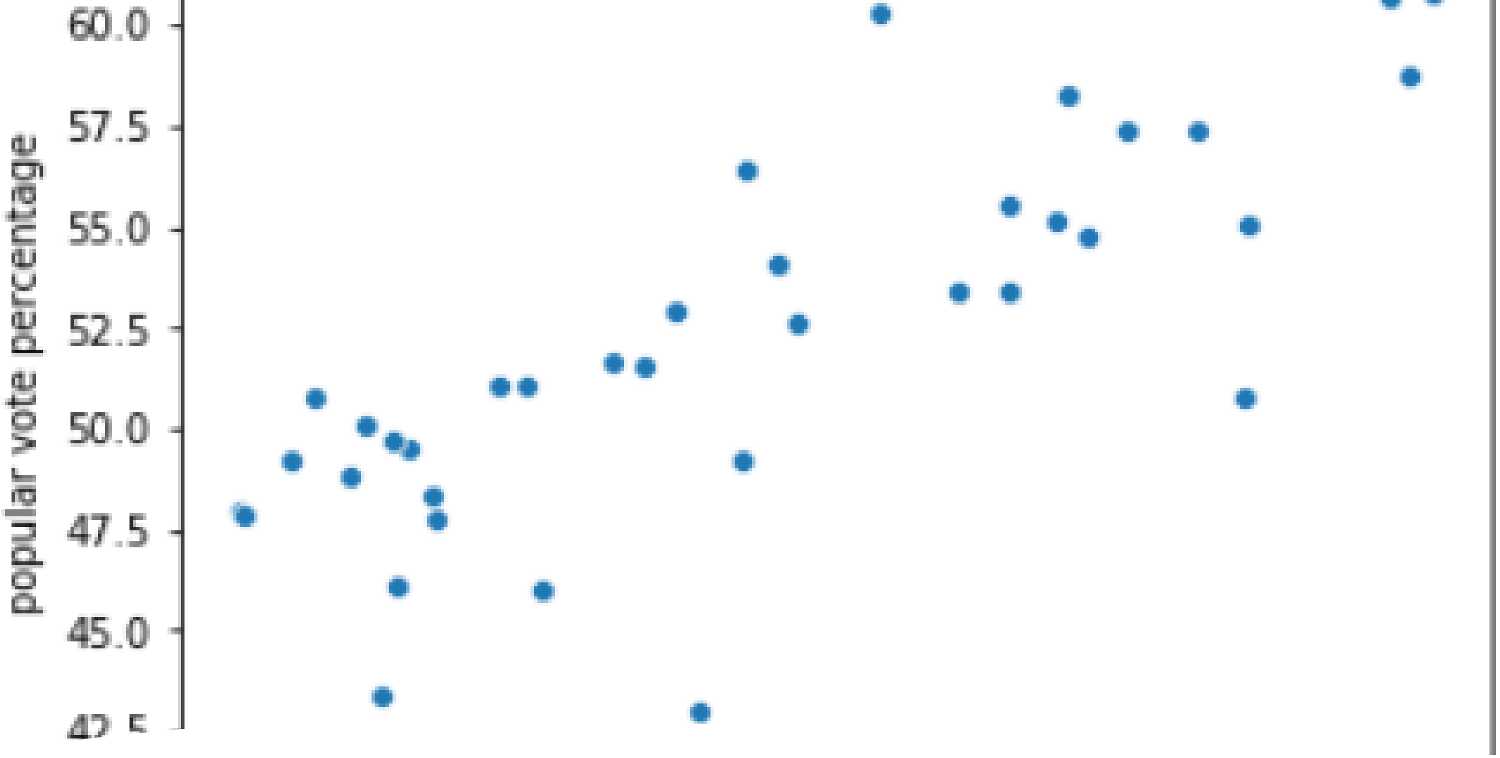
```
df['party'].value_counts().sort_index().plot.barh()
```

Bar Chart



```
df.groupby("party").mean().plot.bar()
```

seaborn



Scatter plot

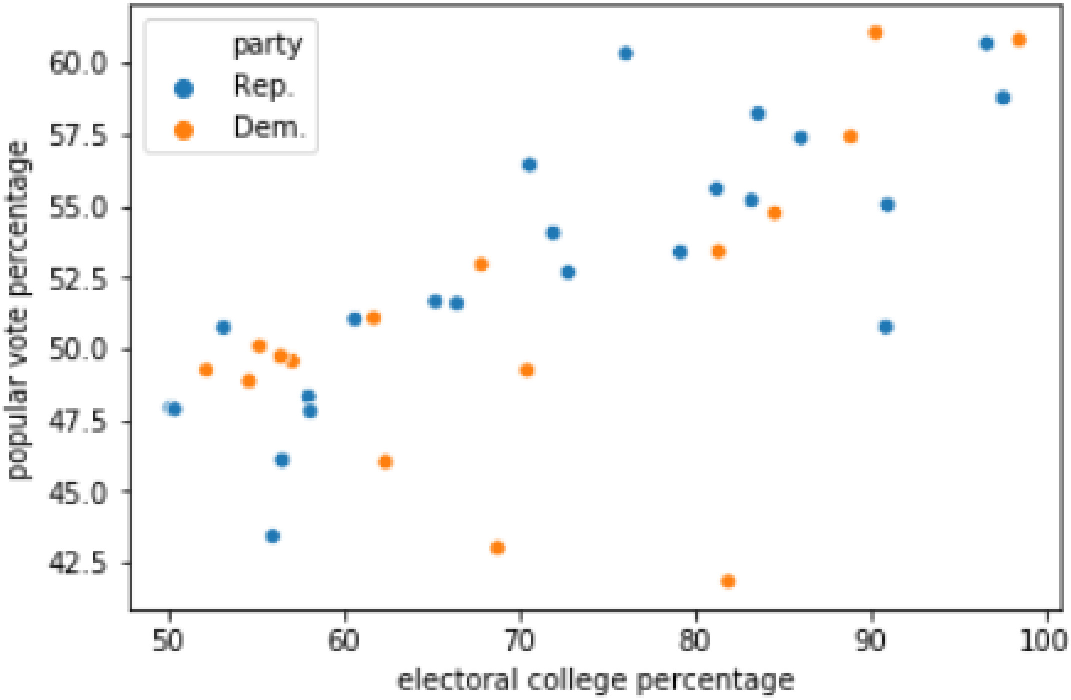
import pandas as pd

import seaborn as sns

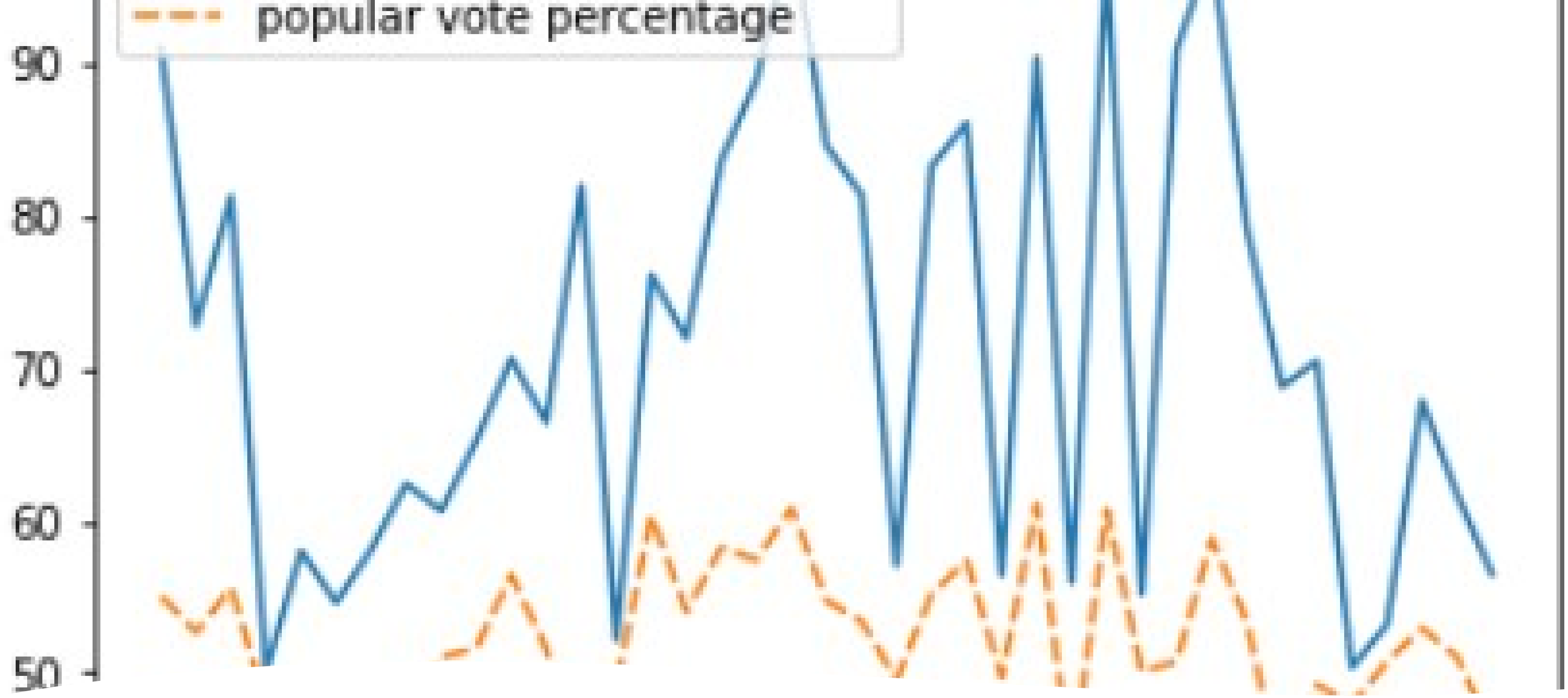
df = pd.read_csv('electionsUSA.csv, sep=";")

sns.scatterplot(x='electoral college percentage', y='popular vote percentage', data=df)

Scatter plot



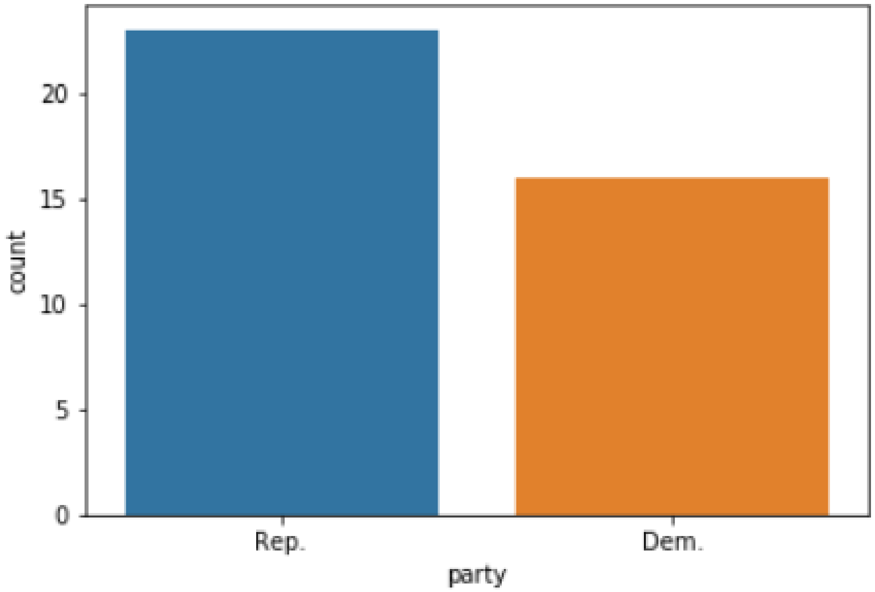
```
import pandas as pd
import seaborn as sns
df = pd.read_csv('electionsUSA.csv, sep=";")
sns.scatterplot(x='electoral college percentage', y='popular vote percentage', hue="party", data=df)
```



Line chart

```
import pandas as pd
import seaborn as sns
df = pd.read_csv('electionsUSA.csv, sep=";")
XY=df[['electoral college percentage','popular vote percentage’]]
sns.lineplot(data=XY)
```

Bar chart



```
import pandas as pd

import seaborn as sns

df = pd.read_csv('electionsUSA.csv, sep=";")

sns.countplot(df['party'])
```

Other charts

- Histograms
- Pairplot (e.g. `sns.pairplot(df)`)
- Heatmaps (e.g. `sns.heatmap(df.corr(), annot=True`)

Conclusions

- Python libraries used in data visualization
- Creating line charts, bar charts and scatter plots using Matplotlib, Pandas and Seaborn